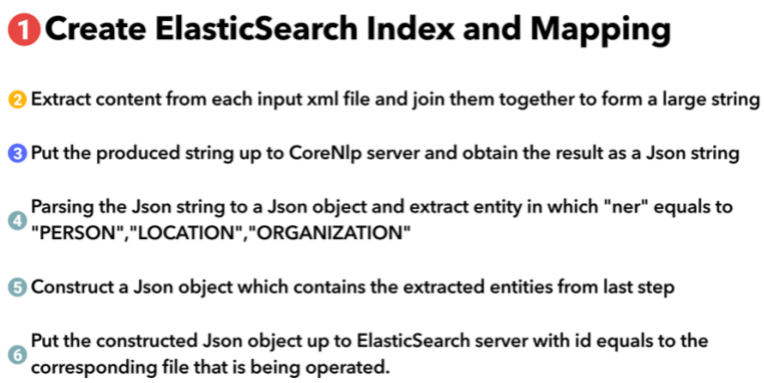
In this report, explanation of index design and solutions to the problem are show. Below is a graph which indicates the procedure of how I complete this assignment:



Firstly, I started coreNlp server using **corenlp-server.sh** command and Elasticsearch server using **elasticsearch** command, then I created a index named “legal_idx” and its mapping:

```
"{"cases":
        {"properties":
                {"name": {"type": "text"},
                "AustLII": {"type": "text"},
                "catchphrases": {"type": "text"},
                "sentences": {"type": "text"},
                "person": {"type": "text"},
                "location": {"type": "text"},
                "organization": {"type": "text"}}
        }
}"
```

Secondly, I loop through each xml file and extract their content, the package I used for reading xml file is **scala.xml.XML** and the method I used for extracting content is “\\”

```
val temp_name = (xmlFile \\ "name" ).text
val temp_Aust = (xmlFile \\ "AustLII" ).text
val temp_catch = (xmlFile \\ "catchphrases").text.trim().split("\n").map(s=>s.trim()).mkString(“ “)
val temp_sen = (xmlFile \\ "sentences" ).text.trim().split("\n").map(s => s.trim()).mkString("  ")
```

I also implement some prepressing on the content I extracted which includes:

1. Trim(): remove unnecessary white spaces at the start and end of the string
2. split("\n"): remove “\n”
3. map(s=>s.trim()): ensure there are no extra spaces
4. mkString: convert them into string

Thirdly, I concatenate all extracted string into one large string and put it up to coreNlp server using **scalaj.http.Http** and obtain the result as a Json string in the format of :

```
{"sentences":
        [{"index":0,"tokens":[
                        {"index":1,"word":"Skymaker","originalText":"Skymaker","lemma":"Skymaker",
                        "characterOffsetBegin":0,"characterOffsetEnd":8,"pos":"NNP","ner":"ORGANI
                        ZATION","before":"","after":" "}…
                        ]
        }]
}
```

Then I implement **JSON.parseFull** method from **scala.util.parsing.json._** package to parse the Json string to a Json object based the following mapping:

```
"{sentence : [ {index:0,tokens:[]}]}": Map[String, List[Map[String, Any]]
```

Then, I extract the “ner” entity from the response, filtering the “ner” entity to obtain text which belongs to “PERSON”, “LOCATION” and “ORGANIZATION“. Moreover, I removed duplicated by using two

MutableList[String], if the text has never appeared in the list, then it will be processed, otherwise if it is in the list, it will be skipped.

After obtaining the desired entities, the next step is to construct a Json object to be used to put up to the Elasticsearch server, the Json object I created has the format like this:

```
val jsonString = s"""{"name":"$temp_name",
                      "AustLII":"$temp_Aust",
                      "catchphrases":"$temp_catch",
                      "sentences":"$sen_modi",
                      "person":"$person",
                      "location":"$location",
                      "organization":"$org"}"""
```

"$XXX" means the variable XXX I created, moreover, I have convert all "\"" in the original text to "\'", as the "\"" will cause some issues when I upload them onto the ElasticSearch server.

The final step is to put the constructed Json object onto the ElasticSearch server.

Example Queries:

Curl command	
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=person:John"	2
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=person:Peter%20Blanas	1
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=location:Melbourne"	1
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=location:Sydney"	1
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=location:Melbourne%20OR%20Sydney"	2
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=organization:Tower%20Software"	2
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=(criminal%20AND%20law)"	2
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=(claims%20AND%20misleading)"	2
curl -X GET "http://localhost:9200/legal_idx/cases/_search?pretty&q=(location:Australia%20AND%20organization:Immigration%20Detention%20Centre)"	1

- The last columns = the number of shards returned
- To run the code, "--packages "org.scalaj:scalaj-http_2.11:2.4.2" is required to be added: spark-submit --packages "org.scalaj:scalaj-http_2.11:2.4.2" --class "CaseIndex" --master local[2] JAR_FILE FULL_PATH_OF_DIRECTORY_WITH_CASE_FILES

In conclusion, I have imported **scala.xml.XML, java.io.File, scala.util.parsing.json._, scalaj.http._, scala.collection.mutable.MutableList** and **"--packages "org.scalaj:scalaj-http_2.11:2.4.2"** is required to be added to spark-submit command to successfully run the code.